Critical Review of Expert System Validation in Transportation

GARY S. SPRING

Expert system validation—that is, testing systems to ascertain whether they achieve acceptable performance levels—has with few exceptions been ad hoc, informal, and of dubious value. Very few efforts have been made in this regard in the transportation area. A discussion of the major issues involved in validating expert systems is provided, as is a review of the work that has been done in this area. The review includes a definition of validation within the context of the overall evaluation process, descriptions and critiques of several approaches to validation, and descriptions of guidelines that have been developed for this purpose.

The subject of expert systems—smart computers solving difficult problems—is intriguing, and the transportation engineering area is ripe for their development. Transportation engineering, perhaps more than other areas of civil engineering, is characterized by experiential problems, such as the design of highway noise barriers, traffic incident management, analysis and design of pavement rehabilitation strategies, and large-scale transportation network design. Automated solutions to such problems offer enormous time savings for very busy, very expensive experts. Furthermore, with the dramatic increases in highway congestion and the increasing recognition that building new roads cannot continue to be the answer to congestion problems (in part because of decreasing economic resources), intelligent transportation improvements have become attractive alternatives to more conventional approaches. Experiential problems, such as those just described, and many of the intelligent improvements lack explicit algorithms and, therefore, until the advent of expert systems, had not been amenable to solution by computer.

Expert systems incorporate human expertise into computer programs that are meant to solve problems requiring a human expert. There are several formal definitions of expert systems, all of which deal in some way with a computer program's ability to solve problems requiring judgment and experience, because of either problem complexity or inadequate input information. Such characteristics cause many of the problems associated with validating expert systems.

During the past 10 years, a large number of prototype expert systems have been developed for a variety of transportation applications (*1–3*). High performance, expert-level computer systems require that the expert system prototype be continuously evaluated during its development. The more the system is used and critiqued, the richer and more refined the system's knowledge becomes. Thus, expert systems must be evaluated as must any software development effort. Yet, as presented in Table 1, only a small percentage (about 20 percent) of prototype systems have been subjected to any type of validation, and only a small percentage of those have undergone a formal evaluation process (*1,3,4*). One major reason for this may lie with the general lack of extended support provided by funding agencies to undergo long-term development efforts. Given the long development time and the large amounts of expert time required for expert systems, this perhaps comes as no surprise. Public agencies are typically not willing to make research commitments for longer than 1 or 2 years—and some do not yet trust the technology. However, technical considerations regarding such questions as proper and effective validation techniques also play an important role in long-term development efforts.

Robust expert systems require huge amounts of expert time over a long period for full implementation. Consequently, investors in such systems expect performance that meets system goals and objectives. For many of the applications to which expert systems are applied, correct outcomes are critical. The consequences of a wrong outcome or decision in a design system could cause substantial costs to be incurred. Indeed, some may even have life and death consequences, such as traffic control systems, design systems for crash cushions, incident management, and so on. In future systems, where the highway system will become increasingly automated, there will be a commensurate increased need for correct decisions.

EXPERT SYSTEM EVALUATION PROCESS

The evaluation process employed for expert systems differs substantially from that used for traditional software projects and is much more problematic, mainly because of the nature of problems solved by the expert system. The process has three major components: verification, validation, and evaluation. Verification may be defined as the demonstration of the consistency, completeness, and correctness of the software (*5*). Software verification determines if the system was built according to specification; its focus, therefore, is on system efficiency. Validation, on the other hand, is concerned with the quality of the conclusions, or solutions to problems, that the software provides; thus, its focus is on system effectiveness. A commonly borrowed phraseology to describe the difference between verification and validation is that verification is concerned with building the system right, whereas validation is concerned with building the right system (*6,7*). Evaluation is concerned with user issues such as user acceptance and system usefulness. It is fairly well agreed that validation is the cornerstone of the system evaluation process and the most difficult component to accomplish.

Since verification of the expert system consists of checking the completeness and consistency of the knowledge base and the quality of the inference engine, it is essentially a programming task and therefore is more amenable to traditional software engineering approaches, which have been well documented by others (*6,8–12*). In contrast, expert system validation has, until recently, been mostly

G. S. Spring, Department of Civil Engineering, North Carolina A&T State University, Greensboro, N.C. 27411.

TABLE 1 Summary of Validation Efforts for Transportation-Related Expert Systems

System Area	*Number Developed*	*Prototype only*	*% in which Validation Attempted*
Design	11	9	18%
Operations	20	15	25%
Maintenance	10	9	10%
Management	17	12	29%

ad hoc, informal, and of dubious value (*6,12*). In recent years, a variety of approaches to validation have been described in the literature. There have also been efforts to provide validation guidelines using one or two of the methods.

The objectives of this paper are to review current validation efforts and to discuss issues related to those efforts as well as solutions to their associated problems. The paper does not provide an exhaustive bibliographic survey of validation activities for knowledge-based expert systems, which would be a daunting task not amenable to the format. It is intended, instead, as an overview of work that has been done in the expert system validation area that will provide an information resource for evaluating and choosing validation techniques for future efforts.

SYSTEM VALIDATION

The definition of where verification ends and validation begins has changed since the earliest validation efforts (*13–15*). These early efforts were concerned primarily with the consistency and completeness of the knowledge base and less with the evaluation of system functioning. Their definitions of validation included detection of problematic rules such as redundant rules, subsumed rules, cyclic rules, and conflicting rules. In recent studies this has been called the logical validation of the system (as opposed to semantic validation) (*7*). Semantic validation includes the examination of the underlying knowledge model that is being implemented; some authors believe that the latter definition fits better with the conceptual definition of validation upon which there exists almost universal agreement—namely, validation is concerned with building the right system (i.e., with meeting the operational goals of the system.) Similar to the differences between precision and accuracy, or management and leadership, problematic rules have more to do with system efficiency than with assessing the quality of the system model. (One can envision a system giving perfectly consistent answers, all of them perfectly wrong.)

Validation Issues

The most common approach to validation is the one used for validating traditional software, namely, outcomes assessment. Other approaches focusing on underlying theoretical concepts are not applicable for most systems since the systems rely on rules of thumb and surface knowledge. Indeed, these other "construct" validation approaches would be impossible for many systems.

The outcomes assessment approach consists of collecting a set of case studies yielding the required coverage for the model being implemented and comparing software outcomes with correct answers. This works well for traditional software (if the cases are chosen properly) because the problems to which it is applied have correct answers. The problems for which expert systems are designed do not have simple right or wrong answers. The correct answer is often a matter of judgement on the part of the human expert (whose judgement is, of course, subject to error). Indeed, there are typically several acceptable alternative solutions that may be used. Thus, the aim of validation is not to determine whether the expert system gives correct answers, but rather whether its answers are valid.

Other issues involving the use of historical case studies include the poor choice of test cases to be used for validation and the potential for human bias for or against computers. With regard to the former, the chosen cases must provide adequate coverage of situations with which the system will be faced. For example, the latter issue consists of two sources of bias: developers and experts. Developers, for example, may bias computer conclusions in the computer's favor (*16*). The use of experts to provide valid answers for validation and to avoid developer bias introduces a new set of problems: experts may be biased against introducing computer systems into their domain, some may not have an adequate level of expertise to be making judgments, and there may not be consistency among the experts used. Biases may be avoided through the use of blind experiments, in which the experts do not know which answers were generated by the computer and which were generated by a human expert. Analytical techniques such as consensus models and the Turing test, described later, have been used to address the competency and consistency issue.

A final issue, determination of level of performance expected for the system, involves establishing a weighting scheme that combines individual weights and the number correctly answered and then finding how high a score must be attained for performance to be considered adequate. Adequate performance level, of course, depends on the application area and the level or risk that is appropriate in that area—a difficult value to quantify. Validation must therefore begin with a definition of system specifications and a set of constraints under which it will operate. It must also include a plan for the stages at which the system will be evaluated. Although, the importance of incorporating validation in the early stages of development has long been recognized (*16*), only recently have studies focused on this issue (*7,17*).

Approaches to Validation

Several approaches to validation are discussed in the literature, a sampling of which is given in Table 2. The methods listed either determine knowledge consistency (logical validation) or quantify the comparisons of system results and expected performance (semantic validation). Wu et al. (*18*) derived rule-dependency graphs to describe the dependency relationships among facts and left-hand and right-hand sides of rules contained in a rule base. Wu maintains that a rule base can be validated by examining the topology of the corresponding rule-dependency graph. The approach detects redundant rules, subsumed rules, cyclic rules, conflicting rules, and unnecessary conditions.

Semantic validity has been measured using a variety of methods, all variations of the simple case study approach, wherein the evaluation criteria consist of a comparison of expert system and human expert conclusions. How closely their conclusions agree is used as a measure of performance. Examples of methods include simple comparisons of case study outcomes (*12,19*), hypothesis tests (*4*), analytical models [for example, the outputs of neural network mod-

TABLE 2 Approaches to Validation

Approach	Validation Level	Notes
Rule Dependency Graphs	Logical	Uses topology of rule base to detect problems
Simple Case Study	Semantic	Uses comparisons of system to historical or field experiments to assess performance
Simulation	Semantic	Uses comparisons of system to simulated results to assess performance
Analytical Models	Semantic	Uses comparisons of system to analytical model results to assess performance
Consensus Models	Semantic	Analytical tools to allow comparisons of system to expert conclusions to assess performance
Turing Tests	Semantic	Analytical tools to allow comparisons of system to expert conclusions to assess performance

els have been compared with those from regression models and from linear discriminant models (*20*)], and simulation models. Simulation model methods are analogous to the simple case study method, in which it is not possible to use case studies; for example, real-time applications such as traffic control systems.

Expert system outcomes are judged for a variety of scenarios. The fact that simulation models are just that—models—must be considered when judging system performance. Simply because the expert system performs well in a simulated environment does not guarantee similar performance in the real world. Prerau et al. (*19*) compare and contrast the verification and validation (V + V) efforts of four very different expert systems. Although the validation approaches used are varied, they all use test cases as the general approach. Table 3 summarizes the characteristics of the four efforts along with several other example systems (*1,4,7*). The authors discuss the advantages and disadvantages of each of the effort's approaches and provide guidance as to when the various approaches are appropriate. For example, the use of historical cases may be helpful where field testing or usage cannot be performed extensively because of the critical nature of the process or the cost of the trials. They further point out that the use of outside experts for validation was valuable in that it also served to validate the project expert's knowledge.

Two powerful analytical techniques for addressing the issues of expert competence and consistency are consensus models and Turing tests. O'Leary and Pincus (*21*) explore the use of consensus models as validation tools. They develop two models of consensus, the first of which is based on the binomial distribution and the second, on Bayesian statistics. The authors provide guidance as to when the models are appropriate and on experimental design questions such as relative competence of experts and how many test cases should be used. They explore the role of consensus as a measure of correctness, indicate how one may consider different levels of expertise in panels of experts used in validation, and attempt to measure consensus among groups of experts regarding expert system performance.

Spring (*4*) used a variant of the Turing test to validate the Hazardous Location Analyst (HLA)—similar to that used for the MYCIN and ONCOCIN systems (*22,23*) and to that recommended by Chandrasekaran for validating medical expert systems (*24*). The test consisted of collecting a set of case studies that had been solved by human experts to be solved by the expert system; and asking expert raters for their qualitative assessments of both sets of conclusions according to a specified rating scheme. In this way the issues relative to using case studies and human resources were addressed. Expert system answers were assessed according to their validity rather than being compared with predetermined correct answers; human bias was avoided since the test's raters are not told which answers are computer and which are human; and a reasonable estimation of adequate performance level was obtained from the raters' assessments of the human expert's performance. An additional benefit offered by this approach was that multiple experts are involved in all assessments, thus avoiding parochial judgments (if the raters are chosen properly) about what constitutes adequate performance.

The analytical techniques used are described by Fleiss (*25*) and have been used to rate laboratories using a panel of expert raters. The techniques allow for an assessment of the consistency among experts—that is, do experts agree among themselves with regard to

TABLE 3 Sample Validation Efforts

System	Domain	Project Expert	Outside Experts	Simulation	Historical Results	Field Experiments	Field Usage
COMPASS	Maintenance of large telephone switching systems	X	X				X
HLA	Diagnosis of high accident highway intersections	X	X		X	X	X
MCES	Process control systems operation & monitoring				X	X	
OA	Plant safety maintenance	X		X	X		
PAMEX	Pavement management	X	X		X	X	
QUAWDS	Human pathologic gait interpretation	X	X		X	X	
SCEPTRE	Pavement management	X			X		
TRANZ	Work zone safety	X	X		X		

both human and computer experts conclusions (and, implicitly, performance levels for each)? If assessments are measured objectively (e.g., experts categorize conclusions about the test cases into levels of performance such as expert, good, fair, or poor), the techniques can be used to test for variation among the raters for both the expert system and human expert.

Automated Approaches to Validation

Although verification is quite advanced (*26*), little work has been done in automating validation—most of which uses automated test case generation. Lee and O'Keefe (*27*) maintain that current automated tools only address logical verification and review several rule-checking programs and other tools such as the Expert Validation Associate (EVA) (*28*). Coenon and Bench-Capon (*29*) developed a rule-based simulation tool, as part of the MAKE project, with the premise that validation of a knowledge base must be carried out in a transparent manner. They postulate that the evaluator must be able to validate system behavior, not just system output. The tool provides the ability to input data interactively, to observe individual inferences without requiring specification of complete test cases, to identify bad rules when incorrect results are obtained, and to determine why expected inferences were not drawn. A potential drawback of the systems involves the focus at the rule level of abstraction. Little attention is given to the big picture, which must be the focus for semantic validation.

Zlatereva (*26*) developed an automated framework for the verification, validation, and refinement of knowledge bases called the VVR system. He defines a knowledge base's theory as being valid if four conditions are met: the theory is (*a*) complete, (*b*) structurally correct, (*c*) consistent, and (*d*) functionally correct (that is, all hypotheses generated agree with semantics of the problem domain). The VVR system reasons about causes of detected structural and functional errors, it helps the user define a set of semantic constraints for the system, and it generates a complete set of test cases to be used in validating the system's knowledge base theory. The VVR has two potential problems: it depends on test cases (discussed earlier) and, since its list of test cases is exhaustive based on the knowledge base, may become unwieldy with large knowledge bases.

French and Hamilton (*17*) developed a verification and validation system called TOP (Terms, Operators, and Production), which applies traditional software verification and validation techniques to expert systems. TOP uses a dynamic test case approach in which system conclusions on test cases are iteratively compared with experts' conclusions. It is designed to be used from the inception of the system to implementation. Similar to the VVR, TOP attempts to provide adequate coverage of the knowledge base by generating cases that fire all rules at least once. Strengths of the two systems include their focus on both logical and semantic validation, aid in rule formulation and coding, and setting of goals, constraints, and specifications for the system. Neither, however, addresses human support issues discussed previously, namely; lack of correct answers, human biases, expert competence, and consistency.

Wentworth et al. (*7*) provide software with their handbook on V + V, which is also meant to be used from inception to implementation. Their approach will be discussed later in the section on V + V guidelines. The handbook also lists several other software packages developed for V + V. The three that were designed specifically for validation are concerned either with logical validation of rules or generation of test cases, not with semantic validation.

GUIDELINES FOR VERIFICATION AND VALIDATION

Many of the development tools for V + V, employing strategies such as those discussed in this paper, as well as others (*30–34*), have focused on a single knowledge-based system. They were not positioned as an integral part of the development environment. This is an important reason that the guidelines discussed here (*7,26,35,36*) and the TOP and VVR systems hold such promise for future developers. They avoid the problems associated with other heretofore fragmentary efforts.

The guidelines follow much of what has been learned about validation over the past 10 years. All state that validation must attempt to measure system effectiveness. Consequently, all provide guidance in setting system goals, constraints, and specifications, and all emphasize the importance of beginning this process at the outset—something which, suprisingly, has not been done very often (*17*).

O'Keefe (*35*) proposes nine guidelines to which, he maintains, any evaluation method should adhere and describes a general multiple-criteria method that fits all of the guidelines and an entreaty to begin evaluation at the start of system development. The multiple-criteria method uses a strategic planning approach that consists of establishing goals, objectives relating to the goals, and measurable criteria relating to each objective. The method then suggests applying the resulting evaluation framework to the existing situation (no expert system), denoting each criterion as an instance of EPS (criterion value prior to system implementation). Expectations for each criterion are then determined, denoted by EE (criterion expected value after system implementation). System performance may then be assessed by examining where on the EPS-EE scale the system lies for each criterion.

Culbert et al. (*36*) developed a method that uses a panel composed of stakeholders in the system—experts, users, developers, and managers—as a review committee for the system. The committee establishes goals, constraints, specifications, and so on for the system at each of the four phases in its life cycle: problem definition, initial prototype, expanded prototype, and delivery/maintenance. The method recommends reasonable goals at each phase. The authors maintain that the primary hindrance to validation is the use of methodologies that do not produce traceable, testable requirements.

Lee and O'Keefe (*27*) present a strategy for V + V composed of establishing criteria, developing a life-cycle model that specifies what V + V step can be done and when, and establishing constraints (and opportunities) imposed by the characteristics of the system being developed. No single strategy is developed because the authors argue that no single strategy can be applied universally.

The VVR and TOP systems reviewed earlier both provide a comprehensive approach to V + V and may be thought of as guidelines as well. They aid in goal formulation and test case generation as well as in the actual coding of rules.

Wentworth et al. (*7*), sponsored by FHWA, developed a handbook for V + V that also attempts to integrate V + V throughout a system's life cycle. The handbook is fairly comprehensive in that it considers logical and semantic validation and is meant to be used as a guide from project inception to implementation. It uses case studies and multiple expert conclusions in its approach to validation, and it includes analytical tools for case study assessment and measuring consistency among experts. Proportions of right answers and the experimental design problems associated with that approach (e.g., size of sample required, estimated statistics for proportions) are also discussed. The handbook is unusual in that it acknowledges the different V + V needs of small systems versus large ones. The validation methodology presented appears simple to follow and implement and is actually provided on disk as well. The user, however, is required to take the authors' word that the techniques presented are indeed the simplest and most powerful. No effort is made to explain why this is so.

CONCLUSIONS

This paper has provided an overview of past and present V + V efforts from which the following conclusions may be drawn:

- Validation should begin at the outset of the development process. There is almost universal agreement on this point. The system's design should facilitate V + V.
- System goals, specifications, and constraints should be well-defined. Again, this facilitates an assessment of system performance.
- An evaluation plan for the system must be established at the outset.
- Most efforts to automate validation have focused on logical validation.
- The system must be subjected to semantic as well as logical validation.
- Expert competency and consistency must be considered in system development and evaluation.
- Analytical models have been used successfully to validate knowledge models.
- Analytical methods are available that will help to avoid problems associated with human support approaches.

The validation of expert systems continues to be less well developed than other aspects of the knowledge engineering process, particularly their semantic validation. However, as the development of expert systems has become more formalized, and judging from the increasing number of validation case studies, a sampling of which are discussed in this paper, more attention is being given to this crucial step in the development process. Inconsistencies in the definitions used for verification, validation, and evaluation remain, but they are decreasing. It is hoped that this discussion will serve as a resource for transportation professionals interested in providing robust, validated expert systems.

REFERENCES

1. Cohn, L. F., and R. A. Harris. *NCHRP Synthesis 183: Knowledge-Based Expert Systems in Transportation*. TRB, National Research Council, Washington, D.C. 1992, 58 pp.
2. Zaman, M., G. Bozai, A. S. Mangaokar, A. Soltani, J. F. Harp, J. L. Gattis, and J. G. Laguros. *Expert Systems in Transportation Engineering: A Critical Review and Application Potential*. Oklahoma Department of Transportation, 1993.
3. Roddis, W. M. K. *Survey of Research in Progress on Expert Systems and Artificial Intelligence in Civil Engineering*. Structural Engineering and Engineering Materials, SL Report 92-2. University of Kansas Center for Research, Inc., 1992
4. Spring, G. S. Validating Expert System Prototypes Using the Turing Test. *Transportation Research C*, Vol. 1, No. 4, 1993, pp. 293–301.
5. Adrion, W., M. Branstad, and J. Cherniavsky. Validation, Verification and Testing of Computer Software. *ACM Computing Surveys*, 1982, pp. 159–192.
6. O'Keefe, R. M., and E. P. Smith. Validating Expert System Performance. *IEEE Expert*, 1987.
7. Wentworth, J. A., R. Knauss, and H. Aougab. Verification, Validation and Evaluation of Expert Systems. In *FHWA Handbook*, 1st ed., 1995, pp. 1–146.
8. O'Keefe, R. M., and D. E. O'Leary. A Review and Survey of Expert System Verification and Validation. *Proc., IEEE Conference on Artificial Intelligence Applications*, 1991.
9. Dym, C. L., and R. E. Levitt. *Knowledge Based Systems in Engineering*. McGraw-Hill, New York, 1990.
10. Gupta, U. *Validating and Verifying Knowledge-Based Systems*. IEEE Computer Society Press, 1990.
11. Waterman, D. A. *A Guide to Expert Systems*. Addison Wesley Publishing Co., 1986.
12. Spring, G. S., J. Collura, P. W. Shuldiner, and J. Watson. Testing, Verification and Validation of Expert Systems: A Case Study. *Journal of Transportation Engineering*, ASCE, Vol. 117, No. 3, 1990, pp. 350–360.
13. Shortliffe, E. H. *Computer-Based Medical Consultations: MYCIN*. Elsevier, New York, 1976.
14. Davis, R. *Use of Meta Level Knowledge in the Construction and Maintenance of Large Knowledge Bases*. Computer Science Department, Stanford University, California, 1976.
15. Suwa, M., A. C. Scott, and E. H. Shortliffe. An Approach to Verifying Completeness and Consistency in a Rule-Based Expert System. *AI Magazine*, 1982, pp. 16–21.
16. Gaschnig, J., P. Klahr, H. Pople, E. Shortliffe, and A. Terry. Evaluation of Expert Systems: Issues and Case Studies. In *Building Expert Systems* (F. Hayes-Roth, D. A. Waterman, and D. B. Lenat, eds.), Addison-Wesley Publishing Company, Inc., Reading, Mass., 1983, pp. 241–280.
17. French, S. W., and D. Hamilton. A Comprehensive Framework for Knowledge-Base Verification and Validation. *International Journal of Intelligent Systems*, Vol. 9, 1994, pp. 809–837.
18. Wu, C. H., S. J. Lee, and H. S. Chou. *Dependency Analysis for Knowledge Validation in Rule-Based Expert Systems*. IEEE, 1994, pp. 327–333.
19. Prerau, D. S., W. L. Papp, R. Bhatnagar, and M. Weintraub. Verification and Validation of Expert Systems: Experience with Four Diverse Systems. *International Journal of Expert Systems*, Vol. 6, No. 2, 1993, pp. 251–269.
20. Tam, K. Y., and M. Y. Kiang. Managerial Applications of Neural Networks: The Case of Bank Failure Predictions. *Management Science*, Vol. 38, No. 7, 1992, pp. 926–947.
21. O'Leary, D. E., and K. V. Pincus. Models of Consensus for Validation of Expert Systems. *International Journal of Expert Systems*, Vol. 5, No. 2, 1993, pp. 237–249.
22. Yu, V. L., et al. Antimicrobial Selection by a Computer. *Journal of the American Medical Association*, 1979, pp. 1279–1282.
23. Hickam, D. H., E. H. Shortliffe, M. B. Bischoff, A. C. Scott, and C. D. Jacobs. The Treatment Advice of a Computer-Based Cancer Chemotherapy Protocol Advisor. *Annals of Internal Medicine*, Vol. 103, No. 6, 1985, pp. 928–936.
24. Chandrasekaran, B. On Evaluating AI Systems for the Medical Diagnosis. *AI Magazine*, 1983, pp. 34–37.
25. Fleiss, J. L. Measuring Nominal Scale Agreement Among Many Raters. *Psychological Bulletin*, Vol. 76, No. 5, 1976, pp. 378–382.
26. Zlatereva, N. P. A Framework for Verification, Validation, and Refinement of Knowledge Bases: The VVR System. *International Journal of Intelligent Systems*, Vol. 9, 1994, pp. 703–737.
27. Lee, S., and R. M. O'Keefe. Developing a Strategy for Expert System Verification and Validation. *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 24, No. 4, 1994.
28. Chang, C. L., J. B. Combs, and R. A. Stachowitz. A Report on the Expert Systems Validation Associate (EVA). *Expert Systems with Applications*, Vol. 1, No. 3, 1990, pp. 217–230.
29. Coenen, F., and T. J. M. Bench-Capon. *KBS Maintenance Validation Using Simulation*. Department of Computer Science, Liverpool University, 1991, pp. 215–228.
30. Nguyen, T. A. Verifying Consistency of Production Systems. *Proc., 3rd IEEE Conference on Artificial Intelligence Applications*, 1985, pp. 375–378.
31. Bahill, A. T., M. Jafar, and R. F. Moller. Tools for Extracting Knowledge and Validating Expert Systems. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol 18, No. 2, 1987, pp. 857–862.

32. Nazareth, D. L., and M. H. Kennedy. Verification of Rule-Based Knowledge Using Directed Graphs. *Knowledge Acquisition*, Vol. 3, 1991, pp. 339–360.
33. Landauer, C. Empirical Methods in Expert System Validation. *Proc., 1986 IEEE International Conference on Systems, Man, and Cybernetics*, 1990, pp. 121–125.
34. Nazareth, D. L., and M. H. Kennedy. Knowledge-Based System Verification, Validation, and Testing: The Evolution of a Discipline. *International Journal of Expert Systems*, Vol. 6, No. 2, 1993, pp. 143–162.
35. O'Keefe, R. The Evaluation of Decision-Aiding Systems: Guidelines and Methods. *North-Holland Information & Management 17*, 1989, pp. 217–226.
36. Culbert, C., G. Riley, and R. T. Savely. An Expert System Development Methodology That Supports Verification and Validation. *ISA Transactions*, Vol. 28, No. 1, 1989, pp. 15–18.

Publication of this paper sponsored by Committee on Artificial Intelligence.